RABBIT: Cognitive Science in Interface Design

by

Michael D. Williams
Frederich N. Tou
Richard E. Fikes
Austin Henderson
Thomas Malone

Cognitive and Instructional Sciences Group
XEROX Palo Alto Research Center
Palo Alto, California

Abstract

A new kind of user interface for information retrieval has been designed and implemented to aid users in formulating a query. The system, called RABBIT, relies upon a new paradigm for *retrieval by reformulation*, based on a psychological theory of human remembering. The paradigm actually evolved from an explicit attempt to design a 'natural' interface which imitated human retreival processes.

To make a query in RABBIT, the user interactively refines partial descriptions of his target item(s) by criticizing successive example (and counterexample) instances that satisfy the current partial description. Instances from the database are presented to the user from a perspective *inferred* from the user's query description and the structure of the knowledge base. Among other things, this constructed perspective reminds users of likely terms to use in their descriptions, enhances their understanding of the meaning of given terms, and prevents them from creating certain classes of semantically improper query descriptions. RABBIT particularily facilitates users who approach a database with only a vague idea of what it is that they want and who thus, need to be guided in the (re)formulation of their queries. RABBIT is also of substantial value to casual users who have limited knowledge of a given database or who must deal with a multitude of databases.

1. Introduction

One way to test a theory is to try to do something useful with it. We have taken a cognitive theory of human remembering together with some artificial intelligence ideas about knowledge representation and used it to design a new paradigm for database retrieval interfaces for casual users. The paradigm is called *retrieval by reformulation.* A small experimental system based on this new paradigm has been implemented in the Smalltalk programming language [Ingalls, 1978] using KloneTalk [Fikes, 1981] on the Xerox Dolphin and Dorado personal computers.

Part of the motivation for designing a new kind of database interface was the unsuitability of existing database interfaces for casual users. Some database interfaces (e.g., SQUARE [Boyce et al, 1975] and SQL [Chamberlin et al, 1976]) require many hours of instruction to learn; others have a syntax which users find difficult to use and understand (e.g., the boolean expressions of DIALOG [Lockheed, 1979]). Interfaces based on the relational data model [Codd, 1970] usually require the user to know in advance which tables and attributes he will be needing, while users of network databases (such as ZOG [Robertson et. al., 1981]) frequently get lost during the course of their search.

To help solve these problems we looked for inspiration to theories of how people retrieve information form their own memory. We believe this approach is promising for two primary reason: (1) To the extent that the interface between a person and his external memory is like the interface between the person and his internal memory the external memory may be easier and more narural to use, and (2) to the extent that human memory systems embody a 'solution' to the problems of retreival from large heterogeneous databases, they may provide useful insights about how to design similar artificial systems.

We began our design process by conjecturing an interface which permitted *descriptive retreival.* The basic tenet of descriptive retrieval is that people retrieve information from (their own) memory by iteratively constructing partial descriptions of the desired target item [Bobrow and Norman, 1975; Norman and Bobrow, 1979; Williams and Hollan, 1981]. The problem was that our conjectured system appeared to give us little more than the traditional boolean expression schemes such as DIALOG. We simply replaced the technical term 'keyword' with the term 'desrciptor.' This led us to a re-examination of the problems inherent in boolean expression interfaces.

Upon consideration we conjectured that there were three major sources of difficulty for casual users of interfaces based upon boolean expressions of keywords: (1) the user has incomplete knowledge about the *descriptive terms* needed to create a query (e.g. what car colors does the database know about? red, crimson, rose, mauve?), (2) the user doesn't know what kind of attributes of the item(s) he is seeking the database recognizes (e.g. does the database even has an attribute for car color?), and (3) many users find the syntax of complex boolean expressions difficult to understand.

Yet, if people actually recall information by descriptive retreival then they must face the same problems; they must have some trick to get by those problems. We found such a trick in *retrieval by instantiation.* Retrieval by instantiation postulates that the information retrieved at each iteration of the retrieval process is often in the form of an *instantiation*, i.e., an example item suggested (e.g., analogically or metaphorically) by the partial description [Williams, 1981]. The common consequence of such an instantiation is that one is 'reminded' of something similar to the original item [Schank, 1980; Kolodner, 1980; Bower, Turner, and Black, 1979]. We conjecture that this reminding serves to counter all three of the problems noted for boolean expression schemes. The instantiations provide a *template* for describing the target item, access to the descriptive terms, and can provide the basis for an incremental reconstruction of the target item that avoids much of the complexity inherent in highly structured descriptions.

2. Retrieval by Reformulation

The basic principle underlying RABBIT is a new paradigm, retrieval by reformulation, for information retrieval elaborated from the notion of retrieval by instantiation. The user makes a query by first constructing a *partial description* of the item(s) in the database for which he is searching. RABBIT then provides a description of an *example instance* from the database which matches the user's partial description. Since it is unlikely that the first instance will be exactly what the user is looking for the user can then select any of the attributes shown in the example and incorporate those descriptors, or variations of them, into his partial description, thus, *reformulating* his initial query. At any time the user can request a new example instance, one which matches the latest version of his (partial) description, and then use the descriptors of that new instance to refine his query description still further.

Figure 1 shows RABBIT in the midst of a retrieval interaction. The interface consists of four primary window panes. The 'Description' pane specifies an implicitly defined boolean expression that appears to the user as a partial description of the item(s) he is seeking. The 'Example' pane contains an example item that matches the partial description as of the last user initiated retrieval cycle from the RABBIT defined perspective. More precisely, it contains a description, called the *image*, of an instance from some well-defined *perspective* (e.g., "STAR 8011 computer" can be viewed from the perspectives of "a manufactured product," "a computer," "an electronic device," "a piece of office equipment," and "a piece of stock in a store."). The 'Matching Examples' pane lists instances which satisfy the partial description as of the last retrieval cycle. The 'Previous Description' pane contains the description used on the last retrieval cycle which determines the perspective for presentation of the example and the list of matching examples. The example pane command pop-up menu is also displayed.

The example instance mentioned above is a central element of the interface. It serves several purposes: it functions as a *template*, it permits *access* to additional descriptors, it provides *semantic resolution* of potentially ambiguous terms, and it frequently serves as a *counterexample.*

The example instance is a template in the sense that its presentation provides a pattern for making a query using the descriptors in the instance's image. It permits access to new descriptive terms through the alternatives and describe commands elaborated below.

It also provides semantic resolution in that the context of a term such as the role name 'manufacturer' establishes and refines the term's meaning. The role name 'manufacturer:' could refer to a person or a nation or a corporation. The statement 'manufacturer: Xerox' in the context of a description of a computer product helps resolve a host of potential meanings.

The example instance is also a counterexample to the user's intentions whenever it is not exactly what the user is looking for. Rather than simply permitting the user to express his displeasure with the counterexample and have RABBIT try to guess what is wrong with it, the system tries to encourage the user to articulate what is wrong with the instance presented. The counterexample's simple presence serves to remind the user that his query description is incomplete or wrong and, in addition, point out the particular parts of his description that need correction or modification.

Finally, since the amount of information known about the retrieved instance could be considerable, the information actually presented in the image is limited to be only that information which can be inferred to be relevant based on the query description the user has given so far. (E.g., information concerning the dinner menu or house specialty of a given restaurant would be available from the perspective of "a place which serves food" but not from the perspective of "a business." So if the user had begun his query with the descriptor '**Business**', then the image of the retrieved instance, even if it is a restaurant, would not, initially, include information about its dinner menu.)

The current implementation of RABBIT supports a small set of 5 basic operations for creating a query description given the descriptors provided in the image of the example instance. These operations, shown in figure 1, are require and prohibit (which specify that the given descriptor is or is not to be a descriptor of the retrieved instance, respectively), alternatives (which presents the user with a popup menu of alternative descriptors to the given one), specialize (which shows the specializations of the given descriptor), and describe (which allows the user to examine a description of a given descriptor or to describe recursively what that descriptor should be). The describe command provides the user with the capability to build *embedded descriptions*, an example of which appears in figure 1 with the value of the attribute 'disk' being itself a description. [Tou, 1982] and [Tou, Williams, Fikes, Henderson, and Malone 1982] contain a more complete discussion of the paradigm of retrieval by reformulation and the user interface to RABBIT.

This paradigm of retrieval by reformulation, in effect, defines a form of interaction by which RABBIT can assist casual users in formulating queries. Much of the intelligence of RABBIT comes from control of this interaction by appealing to the conceptual structure of the database.

3. Perspectives

The KL-ONE epistemology for representing knowledge [Brachman, 1979] has had a major influence on the development of RABBIT. One of the main uses of KL-ONE is the implementation of *perspectives*. A perspective is simply a way of describing an event or item from a particular viewpoint [Bobrow and Norman, 1975, Bobrow and Winograd, 1977, Goldstein and Bobrow, 1980, Goldstein, 1980]. In RABBIT, a perspective specifies which descriptors are included in the image of any instance presented to the user. RABBIT perspectives are dynamic in that the perspective from which the user views the instances in the database changes depending on the current partial description and on where he is within the database.

There are two distinct mechanisms RABBIT uses to construct a perspective. First it filters the attributes to be presented to a user by including only attributes implicitly acknowledged by the user. Since the partial description is a representation of the user's intent to the computer, that description is a legitimate basis for determining what information to include in the image of the example instance. In RABBIT the attributes included in the image are exactly those that belong to the instance classes occurring in the partial description. For example, if one were to see the computer descibed in figure 1 retrieved under the partial description '**Product**' (i.e. without the descriptor '**Computer**') then only the attributes 'name', 'manufacturer', and 'cost' would be presented. Once the user refines the partial description to specify that he is seeking a computer, additional attributes (e.g. 'disk:,' 'cpu:,' ...) would appear.

A second mechanism for creating perspectives actually extends the perspective of any given instance beyond attributes directly held by the object. Note in figure 1 that because the user has created an embedded description about the disk of the computer sought, aspects of the disk that the user considers important (e.g. capacity) have been compressed into the image of the computer.

Perspectives serve four main functions in the RABBIT interface:

--controlling the type and amount of information presented
--facilitating the user's understanding of instances
--enforcing certain kinds of semantic consistency
--organizing and managing heterogeneous data.

4. Summary

This paper has briefly described the process of designing a novel type of database interface named RABBIT. RABBIT relies on a new paradigm for information retrieval, *retrieval by reformulation*, derived from a cognitive science theory of human remembering together with some artificial intelligence ideas about knowledge representation. The four main ideas underlying this paradigm are:

1) retrieval by constructed descriptions
2) interactive construction of queries
3) critique of example instances
4) dynamic perspectives.

The first three of these ideas had their origins in human psychology. The development of the fourth idea—dynamic perspectives—was motivated and influenced strongly by the KL-ONE knowledge representation language.

Cognitive Science has played a crucial role in the design of RABBIT. We take the tentative success of the design as an indication of the potential role of cognitive science in the design of human-computer interfaces.

Acknowledgements

A major portion of this work was carried out by the second author under the auspices of the MIT intern program at Xerox PARC. The authors would also like to acknowledge the original stimulus for this work steming from an exciting conference on artificial intelligence and human-computer interfaces sponsored by the Army Research Institute. In particular, we would like Stan Halpern, Janet Kolodner, and Alan Badre to know a part of what came from their efforts in putting that conference together. We would also like to thank John Seely Brown, Tom Moran, Rick Cattell, Laura Gould, and Richard Burton for their patient discussions and guidance. Each contributed crucial pieces of the puzzle many of which we are still putting together. John Seely Brown's questions in particular guided our pursuit of the use of perspectives. Finally, we would like to thank the other members of the Cognitive and Instructional Sciences Group at Xerox PARC for their continuing support and critique throughout the development of RABBIT.

References

Bobrow, D.G., and Norman, D.A. "Some Principles of Memory Schemata," in D.G. Bobrow and A.M. Collins (Eds.), *Representation and Understanding: Studies in Cognitive Science.* New York: Academic Press, 1975.

Bobrow, D.G., and Winograd, T. "An Overview of KRL: A Knowledge Representation Language," *Cognitive Science*, 1, pp. 3-46, 1977.

Bower, G.H., Black, J.B., and Turner, T.J. Scripts in Text Comprehension and Memory, *Cognitive Psychology*, Vol 1, 177-220. 1979.

Boyce, R.F., Chamberlin, D.D., King, W.F., and Hammer, M.M. "Specifying Queries as Relational Expressions: The SQUARE Data Sublanguage," *Communications of the ACM* **18**, 11 (Nov. 1975), pp. 621-628.

Brachman, R.J., Bobrow, R.J., Cohen, P.R., Klovstad, J.W., Webber, B.L., Woods, W.A. "Research in Natural Language Understanding: Annual Report, 1 September 1978 to 31 August 1979," *BBN Report No. 4274.* Cambridge, MA: Bolt Beranek and Newman Inc., August, 1979.

Chamberlin, D.D., Astrahan, M.M., Eswaran, K.P., Griffiths, P.P., Lorie, R.A., Mehl, J.W., Reisner, P., and Wade, B.W. "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control," *IBM Journal of Research and Development* **20** (Nov. 1976), pp. 560-575.

Codd, E.F. "A Relational Model of Data for Large Shared Data Bases," *Communications of the ACM* **13**, 6 (June 1970), pp. 377-397.

Fikes, R. "Highlights from KloneTalk: Display-Based Editing and Browsing, Decompositions, Qua Concepts, and Active Role-Value Maps," Proceedings of the 1981 KL-ONE Workshop, Jackson, New Hampshire, October, 1981.

Goldstein, I.P. "PIE: A network-based personal information environment." *Proceedings of the Office Semantics Workshop*, Chatham, Mass., June, 1980.

Goldstein, I.P., & Bobrow, D. Descriptions for a programming environment, *Proceedings of the First Annual National Conference on Artificial Intelligence*, Stanford, CA, August, 1980.

Ingalls, D.H. "The Smalltalk-76 Programming System: Design and Implementation," *Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages*, Tucson, AZ.: January 1978, pp. 9-16.

Kolodner, J.L. Retrieval and Organization Strategies in Conceptual Memory: A Computer Model. Research Report #187, Department of Computer Sciecne, Yale University, New Haven, CT. 1980.

Lockheed Information Systems. *Guide to DIALOG Searching*, Palo Alto, CA, 1979.

Norman, D.A., and Bobrow, D.G. "Descriptions: An Intermediate Stage in Memory Retrieval," *Cognitive Psychology* **11** (1979), pp. 107-123.

Robertson, G., McCracken, D., and Newell, A. "The ZOG Approach to Man-Machine Communication," *International Journal of Man-Machine Studies* (1981) 14, pp. 461-488.

Schank, R.C. Failure-driven memory. *Cognition and Brain Theory*, Vol. 1, *4*, 41-60, 1980.

Tou, F. *RABBIT: A novel approach to information retrieval*, unpublished M.S. thesis, Massachusetts Institute of Technology, Cambridge, Mass., forthcoming.

Tou, F.N., Williams, M.D., Malone, T.W., Fikes, R.E., and Henderson, A. RABBIT: an Intelligent Interface. Xerox Technical Report, forthcoming, 1982.

Williams, M.D. "Instantiation: A Data Base Interface for the Novice User," Xerox Palo Alto Research Center Working Paper, 1981.

Williams, M.D., and Hollan, J.D. "The Process of Retrieval from Very Long Term Memory," *Cognitive Science* **5** (1981), pp. 87-119.

Zloof, M.M. "Query by example," in *Proceedings of the National Computer Conference*, AFIPS Press, Arlington, Va., May 1975, pp. 431-437.

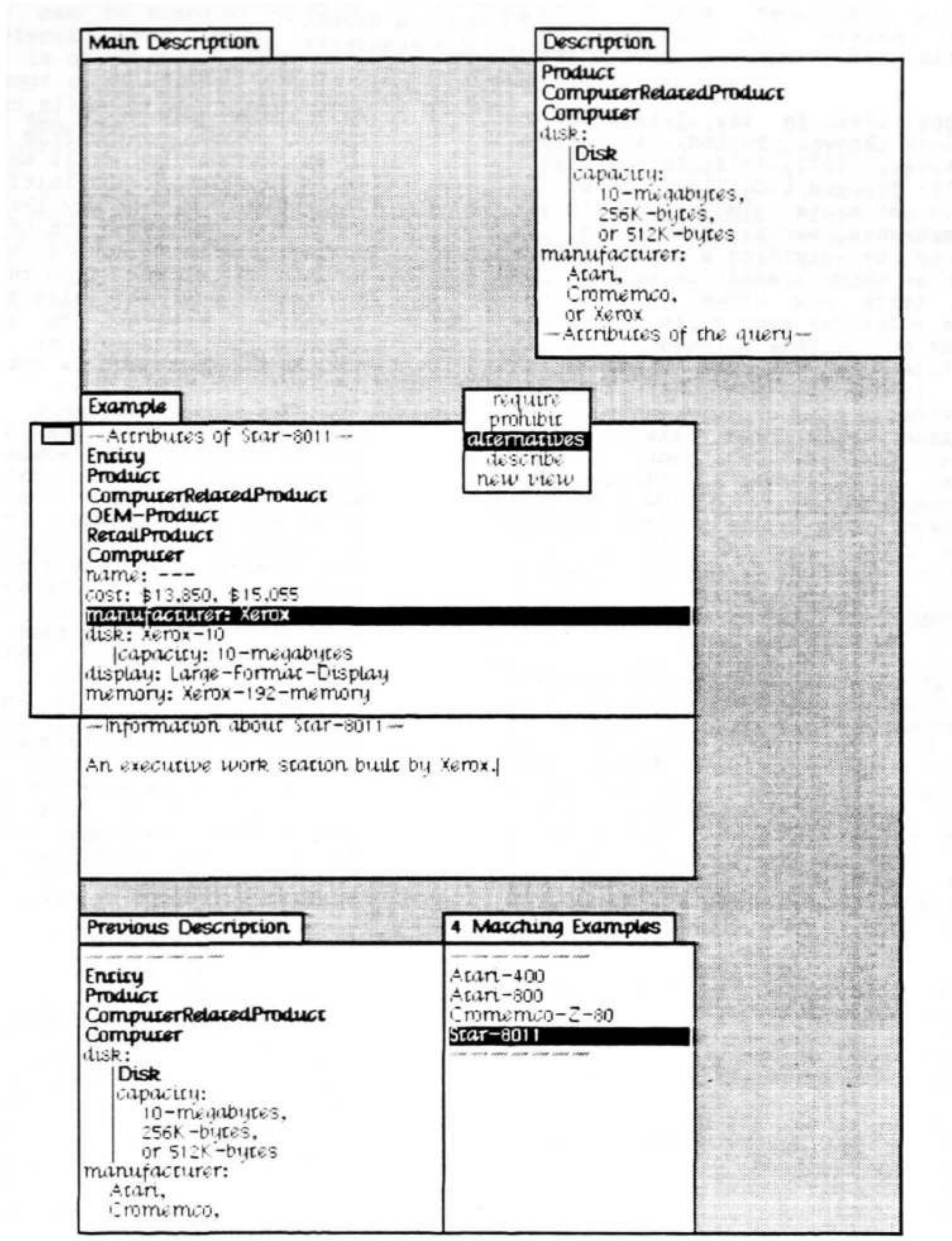

Figure 1. Example of RABBIT display.